2011 Third International Conference on Computational Intelligence, Communication Systems and Networks

Applications of Graph Theory in Computer Science

Ferozuddin Riaz
Computer Science and Engineering Dept
Jubail University College
Jubail Industrial City
Kingdom of Saudi Arabia

Khidir M. Ali
Computer Science and Engineering Dept
Jubail University College
Jubail Industrial City
Kingdom of Saudi Arabia

***Abstract*-Graphs are among the most ubiquitous models of both natural and human-made structures. They can be used to model many types of relations and process dynamics in computer science, physical, biological and social systems. Many problems of practical interest can be represented by graphs. In general graphs theory has a wide range of applications in diverse fields. This paper explores different elements involved in graph theory including graph representations using computer systems and graph-theoretic data structures such as list structure and matrix structure. The emphasis of this paper is on graph applications in computer science. To demonstrate the importance of graph theory in computer science, this article addresses most common applications for graph theory in computer science. These applications are presented especially to project the idea of graph theory and to demonstrate its importance in computer science.**

Keyword–list structure; Matrix structure; Data structure; Algorithm.

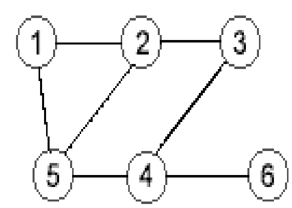
I. INTRODUCTION

In computer science, graph theory is the study of graphs, a mathematical structure used to model pair wise relations between objects from a certain collection. A graph in this context refers to a collection of vertices or nodes and a collection of edges that connect pairs of vertices [1].

Graph theory can be used in research areas of computer science such as data mining, image segmentation, clustering, image capturing, networking etc. Problems of efficiently planning routes for mail delivery, fault diagnostic in computer network and planning a LAN using efficient network topology can be done using graphs. The famous travelling salesman problem asks for the shortest route a travelling salesman [7]. should take to visit a set of cities is a good example of how graph theory might be applied to address some problems in computer science. Graph colouring has a variety of applications to problems involving scheduling and assignments.

Informally, a graph is a set of objects called vertices (or nodes) connected by links called edges (or arcs). Typically, a graph is depicted as a set of dots (i.e., vertices) connected by lines (i.e., edges).

Depending on the applications, edges may or may not have a direction; edges joining a vertex to itself may or may not be allowed, and vertices and/or edges may be assigned weights, that are numbers. If the edges have a direction associated with them (indicated by an arrow in the graphical representation) then it is a directed graph, or digraph**.**



A graph with 6 vertices and 7 edges.

The rest of this paper is organised as follows: section II provides an overview of graph representations using computer systems. In section III graph theoretic data structures are presented. In section IV some of most important graph theory applications are explained. The paper is concluded in section V.

II. GRAPH REPRESENTATIONS USING COMPUTER SYSTEMS

In general, there are four ways to represent a graph in a computer system: The incidence list representation, the incidence matrix representation, adjacency list representation, and the adjacency matrix representation.

Incidence List - This representation uses an array. Each element in the array corresponds with a single edge. Each edge contains a list of size two which corresponds to the endpoints of the edge. This can also apply to directed graphs by ensuring the first vertex be defined as the source or destination while the second should be defined as the opposite.

Incidence matrix - This representation uses a matrix of M edges by N vertices. If the vertex is an

978-0-7695-4482-3/11 $26.00 © 2011 IEEE
DOI 10.1109/CICSyN.2011.40

endpoint to the edge, a value of 1 is assigned to their crossing, otherwise, a value of 0 is assigned. This is a terrible waste of space as every column or row represented by the edge can only have two values of 1 while the rest are labelled 0.

These two representations are most useful when information about edges is more desirable than information about vertices.

Adjacency list - Much like the incidence list, each node has a list of which nodes it is adjacent to. This can sometimes result in "overkill" in an undirected graph as vertex 3 may be in the list for node 2, then node 2 must be in the list for node 3. Either the programmer may choose to use the unneeded space anyway, or he/she may choose to list the adjacency once. This representation is easier to find all the nodes which are connected to a single node, since these are explicitly listed.

Adjacency matrix - there is an N by N matrix, where N is the total number of vertices in the graph. If there is an edge from some vertex x to some vertex y, then the element $M_{x,y}$ would be 1, otherwise it would be 0. This makes it easier to find sub graphs, and to reverse graphs if needed. [6].

These two representations are most useful when information about the vertices is more desirable than information about the edges. Also these two representations are also most popular since information about the vertices is often more desirable than edges in most applications. [7].

III. GRAPH-THEORETIC DATA STRUCTURES

There are different ways to store graphs in a computer system. The data structure used depends on both the graph structure and the algorithm used for manipulating the graph. Theoretically one can distinguish between list and matrix structures but in concrete applications the best structure is often a combination of both. List structures are often preferred for sparse graphs as they have smaller memory requirements. Matrix structures on the other hand provide faster access for some applications but can consume huge amounts of memory. [3].

A. List Structures

There are two types of list structures, which are:

1. Incidence list

The edges are represented by an array containing pairs (tuples if directed) of vertices (that the edge connects) and possibly weight and other data. Vertices connected by an edge are said to be adjacent.

2. Adjacency list

Much like the incidence list, each vertex has a list of which vertices it is adjacent to. This causes redundancy in an undirected graph: for example, if vertices A and B are adjacent, A's adjacency list contains B, while B's list contains A. Adjacency queries are faster, at the cost of extra storage space.

B. Matrix Structures

In incidence matrix the graph is represented by a matrix of size $|V$ (number of vertices) by $|E|$ (number of edges) where the entry [vertex, edge] contains the edge's endpoint data (simplest case: 1 - incident, 0 - not incident).

Adjacency matrix

This is an n by n matrix A, where n is the number of vertices in the graph. If there is an edge from a vertex x to a vertex y, then the element $a_{x,y}$ is 1 (or in general the number of xy edges), otherwise it is 0. In computing, this matrix makes it easy to find sub graphs, and to reverse a directed graph. [7]

Distance matrix

A symmetric n by n matrix D, where n is the number of vertices in the graph. The element $d_{x,y}$ is the length of a shortest path between x and y; if there is no such path $d_{x,y}$ = infinity. It can be derived from powers of A.

IV. GRAPH APPLICATIONS

In computer science, graphs are used to represent networks of communication, data organization, computational devices, the flow of computation, etc. One practical example is the link structure of a website could be represented by a directed graph. The vertices are the web pages available at the website and a directed edge from page A to page B exists if and only if A contains a link to B.

Graph-theoretic methods, in various forms, have proven particularly useful in linguistics, since natural language often lends itself well to discrete structure. Traditionally, syntax and compositional semantics follow tree-based structures, whose expressive power lies in the Principle of Compositionality, modelled in a hierarchical graph. Within lexical semantics, especially as applied to computers, modelling word meaning is easier when a given word is understood in terms of related words; semantic networks are therefore important in computational linguistics. [1].

A graph structure can be extended by assigning a weight to each edge of the graph. Graphs with weights, or weighted graphs, are used to represent structures in which pair wise connections have some numerical values. For example if a graph represents a road network, the weights could represent the length of each road.

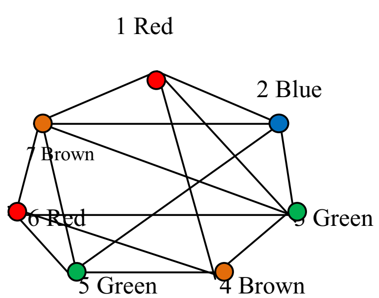
A. Graph Colouring

Colouring of a simple graph is the assignment of a colour to each vertex of the graph so that no two adjacent vertices are assigned the same colour. Graph colouring has a variety of applications to problems involving scheduling and assignments [2]. An example of the graph colouring application is explained below:

Scheduling Examinations

How can the examinations at a university be scheduled so that no student has two exams at[7] the same time? This scheduling problem can be solved using a graph model, with vertices representing courses and with an edge between two vertices if there is a common student in the courses they represent. [7]

Each time slot for exams is represented by a different colour. For instance, suppose there are seven courses to be scheduled. Suppose the courses are numbered 1 through 7. Suppose that following pairs o courses have common students: 1 and 2, 1 and 3, 1 and 4, 1 and 7, 2 and 3, 2 and 4, 2 and 5, 2 and 7, 3 and 4, 3 and 6, 3 and 7, 4 and 5, 4 and 6, 5 and 6, 5 and 7, and 6 and 7. In the figure the graph associated with this set of classes is shown. A scheduling consists of a colouring of this graph. One can verify that the chromatic number of this graph is 4. [7]



(Using colouring to schedule Final Exams)

B. Network Flow Problem

In graph theory, a flow network is a directed graph where each edge has a capacity and each edge receives a flow. The amount of flow on an edge cannot exceed the capacity of the edge. Often in Operations Research, a directed graph is called a network, the vertices are called nodes and the edges are called arcs. A flow must satisfy the restriction that the amount of flow into a node equals the amount of flow out of it, except when it is a source, which has more outgoing flow, or sink, which has more incoming flow. A network can be used to model traffic in a road system, fluids in pipes, currents in an electrical circuit, or anything similar in which something travels through a network of nodes.[10]

The simplest and most common problem using flow networks is to find what is called the maximum flow, which provides the largest possible total flow from the source to the sink in a given graph. There are many other problems which can be solved using max flow algorithms, if they are appropriately modelled as flow networks, such as bipartite matching, the assignment problem and the transportation problem. [10].

C. Algorithms and Graph Theory

The major role of graph theory in computer applications is the development of graph algorithms. Numerous algorithms are used to solve problems that are modelled in the form of graphs. These algorithms are used to solve the graph theoretical concepts which in turn used to solve the corresponding computer science application problems. [4]Route problems can be used to find the shortest route and according a well defined algorithm can be used for such type of a problems and finally these statements can be converted to programming code to find the computer based solution of problem.[5] Some of the well known algorithms are as follows:

1. Shortest path algorithm in a network .
2. Finding a minimum spanning tree.
3. Finding graph planarity.
4. Algorithms to find adjacency matrices.
5. Algorithms to find the connectedness.
6. Algorithms to find Hamiltonian path and the cycles in a graph. [7].
7. Algorithms for searching an element in a data structure (DFS, BFS).
8. Route inspection problem (also called the "Chinese Postman Problem").
9. Travelling salesman problem.

Various computer languages are used to support the graph theory concepts. [5]The main goal of such languages is to enable the user to formulate operations on graphs in a compact and natural manner some graph theoretic languages are:

1. SPANTREE – To find a spanning tree in the given graph.
2. GTPL – Graph Theoretic Language
3. GASP – Graph Algorithm Software Package
4. HINT – Extension of LISP
5. GRASPE – Extension of LISP
6. IGTS – Extension of FORTRAN

Graph modelling language (GML) is a hierarchical ASCII-based file format for describing graphs. It has been also named Graph Meta Language.

Applications supporting GML are:

1. Clairlib, a suite of open-source Perl modules intended to simplify a number of generic tasks in natural language processing (NLP), information retrieval (IR), and network analysis (NA).
2. Cytoscape, an open source bioinformatics software platform for visualizing molecular interaction networks, loads and save previously-constructed interaction networks in GML.

3. NetworkX, an open source Python library for studying complex graphs.
4. Ocamlgraph, a graph library for OCaml.
5. OGDF, the Open Graph Drawing Framework, an open source C++ library containing implementations of various graph drawing algorithms. The library is self contained; optionally, additional packages like LP-solvers are required for some implementations.
6. Tulip (software) is free software in the domain of information visualisation capable of manipulating huge graphs (with more than 1.000.000 elements).

C. Map Colouring and GSM Mobile Phone Networks

GSM is a mobile phone network where the geographical area of this network is divided into hexagonal regions or cells. Each cell has a communication tower which connects with mobile phones within the cell. All mobile phones connect to the GSM network by searching for cells in the neighbours. Since GSM operate only in four different frequency ranges, it is clear by the concept of graph theory that only four colours can be used to colour the cellular regions. These four different colours are used for proper colouring of the regions. Therefore, the vertex colouring algorithm may be used to assign at most four different frequencies for any GSM mobile phone network.[8] Given a map drawn on the plane or on the surface of a sphere, the four colour theorem states that it is always possible to colour the regions of a map properly using at most four distinct colours such that no two adjacent regions are assigned the same colour.

D. Graph Algorithm in Computer Network Security

The vertex cover algorithm (Given as input a simple graph G with n vertices labelled 1, 2, …, n, search for a vertex cover of size at most k. At each stage, if the vertex cover obtained the size at most k, then stop. Simulate the propagation of stealth worms on large computer networks and design optimal strategies to protect the network against virus attacks in real time. Simulation is carried out in large internet like virtual network and showed that the topology routing has big impact on worm propagation [9]. The importance of finding the worm propagation is to hinder them in real time. The main idea here is to find a minimum vertex cover in the graph whose vertices are the routing servers and the edges are the connections between the routing servers. Then an optimal solution is found for worm propagation and a network defence strategy is defined. In a graph G, a set of edges g is said to cover G if every vertex in G is incident on at least one edge in g. The set of edges that covers a graph G is said to be an edge covering or a covering sub graph or simply a covering of G. A spanning tree in a connected graph is a covering. A Hamiltonian circuit is also a covering. [8].

F. Graph Theory and Ad-Hoc Networks

Discuss the role of graph theory related to the issues in Mobile ad-hoc Networks (MANETS)[10]. In Adhoc networks, issues such as connectivity, scalability, routing, modelling the network and simulation are to be considered. Since a network can be modelled as a graph, the model can be used to analyze these issues. Graphs can be algebraically represented as matrices. Also, networks can be automated by means of algorithms. The issues such as node density, mobility among the nodes, link formation between the nodes and packet routing have to be simulated. To simulate these concepts random graph theory is used. The connectivity issues are analyzed by using graph spanners, (A geometric spanner or a *k*-spanner graph or a *k*-spanner). [3].

V. CONCLUSION

This paper has presented applications of graph theory in computer science.

The main aim of this paper is to present the importance of graph theory principles and ideas in various areas of computer applications so researches can apply graph theory to model and optimise different processes in computer science field. An overview of most common applications of graph theory is illustrated. These applications are presented especially to project the idea of graph theory and to demonstrate its importance in computer science.

REFERENCES

[1] N. Biggs, E. Lloyd, and R. Wilson, "Graph Theory,1736-1936, Oxford University Press, 1986.

[2] N. Robertson, D. Sanders, P. Seymour and R.Thomas,"The four color theorem", Journal of Combinatorial Theory Series B **70**: 2–44, 1997.

[3] J. Bondy,. A. Murty, "Graph Theory, Springer", ISBN 978-1-84628-969-9.U.S.R., (2008).

[4] G. Alan "Algorithmic Graph Theory", Cambridge University Press, 1985.

[5] G. Martin "Algorithmic Graph Theory and Perfect Graphs," Academic Press, 1980.

[6] H. Frank, "Graph Theory", Reading, MA, Addison-Wesley, 1969.

[7] K. Rosen, "Discrete Mathematics and its Application", 6th Ed., McGraw-Hill.

[8] J. Pal, S. Sarma and B. Ray, "CCTP, Graph Coloring algorithms – Soft computing Solutions", IEEE, 2007.

[9] N. Deo, "Graph theory with applications to engineering and computer science", Prentice Hall of India, 1999.